Agent–Based Network Infrastructure for E–Communities

M. Mari
Università degli Studi di Parma, Italy

A. Poggi
Università degli Studi di Parma, Italy

M. Tomaiuolo
Università degli Studi di Parma, Italy

INTRODUCTION

Virtual communities represent the gathering of people, in an online “space” where they come, communicate, connect, and get to know each other better over time. Virtual communities can be created for different purposes moving from the simple socialization among people to the collaboration among remote people working on shared projects.

Virtual communities are usually managed through dedicated software systems, also called social software systems, providing the basilar members management services and a set of services specialized for the kind of members they connect. The realization of efficient and reliable systems for the management of virtual communities is not easy because they need to manage the interaction among remote users, possibly connected through heterogeneous networks and often requiring services that cannot be managed through the classical Web-based solutions because sometimes these services require to access software/hardware resources available at the user site.

Virtual communities are usually managed through dedicated software systems, also called social software systems, providing the basilar members management services and a set of services specialized for the kind of members they connect. The realization of efficient and reliable systems for the management of virtual communities is not easy because they need to manage the interaction among remote users, possibly connected through heterogeneous networks and often requiring services that cannot be managed through the classical Web-based solutions because sometimes these services require to access software/hardware resources available at the user site.

Agent technology has been proven as one of the most interesting technologies to simplify the interaction among remote users and applications. In fact, it is well-suited for applications that are communication-centric, based on distributed computational and information systems, and requiring autonomous components readily adaptable to changes. Moreover, the use of agent technology for realizing complex systems provides conceptual simplicity, enhances scalability, and makes interactions in a large collection of information sources and users become tractable.

As introduced in Hattori et al. (1999), there are several characteristics of network communities that make the use of multi-agent systems attractive to realize support systems for managing the interactions among the members of the communities. First, the participants of a network community are widely distributed, and the number of potential participants can be large. Another characteristic is that communities have a dynamic nature because the active members will change over time, in addition to roles of individuals and objectives, and moreover, the community will likely change its aspect. In other words, there does not exist a fixed organization nor a clear goal for a network community. This characteristic contrasts the area of groupware, which helps people already organized to work cooperatively where the members, their roles, and their objectives are rather clearly defined. In addition, the individuality of each member is preserved. That is, each member can have diverse objectives, even if all members share common interests in general. Furthermore, people can be members of several communities at the same time, depending on their various interests. Hence, support needs to be personalized to adapt individual objectives and interests. They also need to adapt to the variations and changes of interests and activities of individuals.

In particular, RAVE is a system for realizing agent-based applications that provide information and expert searching facilities for communities of users working or interested in common or similar topics. The main feature of such a system is the use of multi-agent technologies for realizing a network infrastructure for virtual communities support that can be easily customized to provide services specialized for specific kinds of interests/topics.
Agent-Based Network Infrastructure for E-Communities

BACKGROUND

Several prior systems support expertise recommendations in virtual communities and organizations. Expert Finder is a system that recommends individuals who are likely to have expertise in Java programming (Vivacqua & Lieberman, 2000). This system analyzes Java code and creates user profiles based on a model of significant features in the Java programming language and class libraries written by the user. User profiles are then used to assist novice users in finding experts by matching her/his queries with user profiles. Expertise Recommender (ER) is another system that recommends people who are likely to have expertise in a specific area (McDonald, 2003). A user gathers recommendation from ER by picking a relevant identification heuristic, selecting a matching technique, and entering a description or terms related to a problem. Then, the system responds with a list of individuals who are likely to have expertise with the problem and who are a good social match for the person making the request. In this system, user profiles are built by processing user’s day-to-day work products. MARS is a referral system based on the idea of social network (Yu & Singh, 2003). This system is fully distributed and includes agents who preserve the privacy and autonomy of their users. These agents build a social network learning models of each other in terms of expertise (ability to produce correct domain answers), and sociability (ability to produce accurate referrals), and take advantage of the information derived from such a social network for helping their users to find other users on the basis of their interests.

Several prior systems use multi-agent systems to provide services for virtual communities and organizations. CASBA is a system providing electronic commerce services in virtual communities (Kraft et al., 2000). In particular, this work integrates a multi-agent market framework into a virtual community support system, called VR-SHOP that provides 3-D graphical representations of the participants to provide a more natural interaction among them. TeleCare is a configurable framework for virtual communities focused on supporting assistance to elderly people (Camarinha-Matos & Afsarmanesh, 2005). TeleCare provides, besides classical services for virtual community management, a set of services specialized for the elderly, that is, specialized user interfaces usable by people that are not familiar with computers, living status monitoring, agenda reminder and entertainment.

The main feature of such a system is the integration of multi-agent and federated information management approaches to provide a flexible infrastructure on top of which realize a set of specialized care services.

RAVE

Remote assistance virtual environment (RAVE) is a system for realizing agent-based applications that provide information and expert searching facilities for communities of users working or interested in common or similar topics.

A RAVE-based system associates a personal assistants with each user which helps her/him to solve problems and find relevant information: the assistance is provided proposing information extracted from some information repositories and getting its user in touch with “experts” on the topic, selected on the basis of their profile. A personal assistant also maintains a user profile centered on her/his competences and experience. The profile is built through the positive answers given to other users and by extracting information form the documents that the user produced or collected.

A RAVE system is given a dynamic network of peers, each of them represented by an agent platform, distributed on one or more computing nodes and containing, besides the agent assistants, some agents for managing the access to the information maintained in such computing nodes.

The main feature of this system is that it can be easily customized to manage different services to support the exchange of information among the members of a community. These services can be oriented to support the direct interaction among the members and/or to support the access to the information produced or collected by the different users of a community. Moreover, some services can be useful for any kind of information; other services are specific to a specific type of information (e.g., java programming, photo collections, etc.). In particular, while the interaction between agents and between an agent and its user is independent from the type of services offered to the community, domain ontologies and terms used to realize user and document profile are, of course, different. This last fact does not affect the algorithms used for retrieving the information (that are the same for all the services), but, for example, suggests the realization of specialized user interfaces for helping in the interaction about a specific interest/topic.
System Agents

RAVE provides seven different kinds of agents: access managers, personal assistants, user profile managers, document managers, e-mail managers, and directory facilitators.

Access managers are responsible for activating a personal assistant when a user logs on and for deactivating it when its user logs off.

Personal assistants are the agents that allow the interaction between the user and the different parts of the system and, in particular, between the users themselves. Moreover, these agents are responsible for building the user profile and maintaining it when their user is “online”. User-agent interaction can be performed in two different ways: when the user is active in the system, through a graphical user interface (usually a Web interface); when it is “off-line” through e-mails. In a RAVE system there is a personal assistant for each online user, but, when needed, personal assistants are created to interact with “off-line” users via e-mails.

User profile managers are responsible for maintaining the profile of “off-line” users and of activating personal assistants when it is necessary that they interact with their “off-line” users via e-mails.

Document (and information) managers are responsible of maintaining the pieces of information/documents contained into the system and of finding the appropriate information to answer the queries done by the users connected to the system. In a system there can be different information managers either because they can manage different kinds of information or because they manage information of different groups of users. Moreover, in some cases the distribution and/or the amount of information to be managed requires the introduction of more than an information manager.

E-mail managers are responsible for sending/receiving e-mails to/from “off-line” users. When an e-mail is received from an “off-line” user, they are responsible for activating the corresponding personal assistant and forwarding it the e-mail.

Proxy assistants are responsible for allowing the interaction between a user and her/his personal assistant when the user is out of the office and cannot directly interact with the RAVE peer maintaining her/his personal assistant, for example, because that peer does not provide the access via a Web browser outside the company intranet. In this case the user can be hosted by another RAVE peer that provides her/him a proxy assistant with the only capability of interconnection the user with her/his personal assistant.

Directory facilitators are responsible to inform an agent about the address of the other agents active in the system (e.g., a personal assistant can ask about the address of all the other personal assistants, of the user profile managers, etc.).

User and Document Profile Management

In order to simplify, speed up and reduce the possibility of inaccuracy due to people’s opinions of themselves and to incomplete information, both user and document profiles are computed in an automated way. User and document profiles are represented by vectors of weighted terms whose value are related to the frequency of the term in the document or the frequency of the term in a set of documents associated with the user. In particular, base profiles are computed by using term frequency inverse document frequency (TF-IDF) and the value associated with each term corresponds to the TF-IDF weight (Salton, 1989).

There may be different user and document profiles for the different “services” provided by the community support system. For example, the set of terms used in the profiles can be directly extracted from the documents shared in the community, can be defined by some community glossaries or can be the terms associated with
Agent-Based Network Infrastructure for E-Communities

the different concepts and roles of some community ontologies. Moreover, the user profile may be built either on the basis of the documents she/he wrote or on the basis of the documents she/he collected.

To take into account the opinions of the users on the documents they retrieved and the answers they received from the other users, documents and users have an additional rating profile, containing the same terms of their TF-IDF profile and documents and users are selected on the basis of a dynamic profile built multiplying the TF-IDF profile with the rating profile.

While TF-IDF profiles are periodically updated to take into account the introduction of new documents, rating profile terms have a weight between 0 and 1, with an initial weight of 0.5, and are updated each time a document is retrieved and then rated by a user or when a user’s answer is received and rated by another user.

Some problems have risen applying this profiling approach in our multi-platform and distributed system. In fact, TF-IDF algorithm is generally used only in centralized systems because the relevance of a term should be computed on all the information managed by the system. A similar centralized approach could be realized in a distributed system, but it is quite impossible to be realized for a system similar to RAVE, where system nodes (and so the information available) vary over the time. When a user retrieves information from different RAVE peers, the rating of the information related to users and documents profiles built through the TF-IDF algorithm on a local system can cause two types of errors: the retrieved information may not be the most relevant on the RAVE network and it might not be possible to order on the basis of their relevance information coming from different peers (i.e., their rating was computed on different set of information). Of course, different algorithms can be applied to reduce the problems caused by this kind of error, however, the more the algorithm is sophisticated the better are the results, but, of course, worse are the performances and so the user can wait for the results for a long time. To avoid this last problem, RAVE uses local TF-IDF-based profiling, but tries to get the most relevant information asking a different number of information items to each information manager. This number is calculated on the basis of a rating profile built on the basis of the rating profiles maintained by this information manager: therefore, a personal assistant requests a greater number of documents to the information manager whose users and documents gave the best answers to the other users of the community on the topics represented by the same terms contained in the current query of this personal assistant. Using this solution, there is not the possibility of ordering information items coming from different peers on the basis of their relevance information; therefore, personal assistants order information getting items in a round-robin way from the list coming from the different peers starting, of course, from the most valued ones.

System Security

The information stored into the different repositories of a RAVE system is not accessible to all the users of the system in the same way because it is important to avoid the access to private documents and personal files. RAVE takes care of users’ privacy allowing the access to the information on the basis of the identity, the roles and the attributes of the querying user.

Of course, different levels of privacy can be assigned to the information stored into the same repository. Various models exist to deal with the authorization problem (Sandhu & Samarati, 1994).

The best known is the discretionary access control (DAC) model. It is the traditional model, based on Access Control Lists. In this model, each user is associated with a list of granted access rights. On the basis of this list of permissions, he will be allowed or denied access to a particular resource. A resource administrator is responsible for editing the Access Control Lists.

Another popular model is the mandatory access control (MAC), used to implement multilevel secure (MLS) systems. In these systems, each resource is labeled according to a security classification. Correspondingly, each principal is assigned a clearance, which is associated with a classification list. This list contains all the types of resources the principal should be allowed to access, depending on their classification. The multilevel security is particularly popular in the military field, and in inherently hierarchical organizations.

Another interesting model is the role-based access control (RBAC) model. This model is centered around a set of roles. Each role can be granted a set of permissions, and each user can be assigned to one or more roles. A many-to-many relationship binds principals and the roles they are assigned to. In the same way, a many-to-many relationship binds permissions and the roles they are granted to, thus creating a level of indi-
rection between a principal and his access rights. This also leads to a better separation of duties (between the assignment of principals to roles and the definition of role permissions), to implement privilege inheritance schemes among superior and subordinate roles and to permit temporary delegations of some of the assigned roles towards other principals.

Following the RBAC model, each resource manager of a RAVE system (i.e., each node in the peer-to-peer network) has to deal with three main concepts: principals (i.e., authenticable entities which act as users of resources and services), permissions (i.e., rights to access resources or use services) and roles. The fundamental principle here is that each node is in charge of defining its own roles, and of assigning principals to them.

In a RAVE system, authentication and authorization are performed on the basis of the local knowledge base of trusted users, though they can be delegated to external entities through an explicit, certificate-based delegation (Ellison, 1999). In this sense, the system completely adheres to the principles of trust management. The definition of roles and attributes is also made in a local namespace, and the whole system is, in this regard, completely distributed. Local names are distinguished by prefixing them with the principal defining them, that is, a hash of the public key associated with the local runtime. Links among different local namespaces, again, can be explicitly defined by issuing appropriate certificates.

In this sense, local names are the distributed counterpart of roles in Role-Based Access Control frameworks (Li & Mitchell, 2003). Like roles, local names can be used as a level of indirection between principals and permissions. Both a local name and role represent at the same time a set of principals, as well as a set of permissions granted to those principals. But, while roles are usually defined in a centralized fashion by a system administrator, local names, instead, are fully decentralized. This way, they better scale to internet-wide, peer-to-peer applications, without loosening in any way the principles of trust management.

System Behavior

A quite complete description of the behavior of a RAVE system can be given showing the scenario where a user makes a query to its personal assistant about a particular problem and the personal assistant finds some information and/or some experts that may help her/him. The description of this scenario can be divided in the following steps: query building, query management, answers reception and answers rating.

- **Query building:** The user builds the query through a user interface that, depending on the particular service requested, allows her/him to write a textual query or help and constrain her/him in the form of the query and in the term used. Then she/he forwards the query to her/his personal assistant.

- **Query management:** The personal assistant performs different actions and interacts with different agents to collect the various types of answers.

  For getting information from some information repositories, the personal assistant asks the directory facilitator about all the information managers maintaining such type of information (i.e., information managers maintaining software documentation, e-mails archives, scientific publications, etc.). After receiving this information, the personal assistant forwards the query to all these agents. These agents search information related to the query, associate a score with each information then send them to the personal assistant.

  More complex is the reception of answers directly from some community experts and this step can be divided in four further steps: query submission, experts rating, expert selection and answers submission.

  - **Query submission:** The personal assistant asks the directory facilitator about the other active personal assistants (i.e., the personal assistants of the users that are “online”) and all the user profile managers of the system (i.e., the agents managing the profile of the users that are not “online”). After receiving this information, the personal assistant forwards the query to those agents.

  - **Experts rating:** All the agents that received the query evaluate the capability of their users to answer to this query on the basis of the query itself and of the user profile. The agents that compute a positive score (i.e., its user may give an appropriate answer to the query) reply to the querying personal assistant with the rating of their users.

  - **Experts selection:** The personal assistant divides online and off-line users in two separated lists, orders them on the basis of their rating and, finally,
presented them to its user. The user can select more than one user and then the personal assistant sends the query to the corresponding personal assistants (for the “online” users) and to the corresponding user profile managers (for the “off-line” users).

- **Answers submission:** The replying personal assistants immediately present the query to their user and forward the answers as soon as their user provides it. More complex is the submission of the answers coming from the “off-line” users. In this case, the user profile managers activate the personal assistants of the involved users. These personal assistants forward the query to their user via e-mail and then terminate themselves. Users can answer either via e-mail or when they login again to system. In the case of e-mail, the e-mail manager starts the appropriate personal assistant that extracts the answer from the e-mail and forwards it to the querying personal assistant.

- **Answer reception:** After the reception of all the answers, or when the deadline for sending them expired, the personal assistant orders them and then presents them to its user.

- **Answers rating:** After the reception of the answers, the user starts their evaluation and can inform her/his personal assistant about the rating of some of them. After the rating, the agent forwards each rating to the appropriate personal assistant (for “online” users), user profile manager (for “off-line” users) or information manager that consequently provides to update the corresponding user/information profile.

**System Services**

As introduced below, a RAVE system can be specialized to provide different services. Up to now, RAVE has been used to provide a general service, called Remote Assistant for Information Sharing (RAIS), for providing document sharing and finding “experts” on any kind of information stored in a community, and a specialized service, called Remote Assistant for Programmers (RAP), for supporting communities of students and programmers during shared and personal projects based on the use of the Java programming language.

While the first service is given free by the RAVE system, the second required some work for the management of the profiles and for realizing a user interface that simplifies the preparation of queries. In particular, the profiles are built on the basis of a well-defined set of terms based on a Java programming glossary (in the first experimentations it was used the Java glossary available from Sun Web site (Sun, 2004)) and containing the names of the classes and methods of the Java software libraries shared in their projects by the members of the community.

Both services were experimented by some users. In particular, RAP has been tested in some practical courses on Java where a part of students work was the realization of some Java programs; RAIS has been tested...
among some researchers and students for exchanging information about common research/study interests.

System Implementation

RAVE has been designed and implemented taking advantage of agent, peer-to-peer, information retrieval and security management technologies and, in particular, of two main software components: JADE (Bellifemine et al., 2001; JADE, 2006) and JXTA (Gong, 2001; JXTA, 2005).

JADE (Java Agent Development Framework) is probably the most known agent development environment enabling the integration of agents and both knowledge and Internet-oriented technologies. JADE allows building agent systems for the management of networked information resources in compliance with the FIPA specification (FIPA, 2006). JADE provides a middleware for the development and execution of agent-based applications which can make work seamless and interoperate both in wired and wireless environment. Moreover, JADE supports the development of multi-agent systems through the predefined programmable and extensible agent model and a set of management and testing tools. Currently, JADE is considered the reference implementation of the FIPA specifications and is one of the most used and promising agent development frameworks. In fact, it is available under an LPGL open source license, it has a large user group, involving more than two-thousand active members, it has been used to realize real systems in different application sectors, and its future development is guided by a governing board involving some important industrial companies.

The JADE development environment does not provide any support for the realization of real peer-to-peer systems because it only provides the possibility of federating different agent platforms through a hierarchical organization of the platform directory facilitators on the basis of a priori knowledge of the agent platforms addresses. Therefore, we extended the JADE directory facilitator to realize real peer-to-peer agent platforms networks thanks to the JXTA technology (JXTA, 2006) and thanks to two preliminary FIPA specifications for the Agent Discovery Service (FIPA, 2003) and for the JXTA Discovery Middleware (FIPA, 2004).

JXTA technology is a set of open, general-purpose protocols that allow any connected device on the network (from cell phones to laptops and servers) to communicate and collaborate in a peer-to-peer fashion. The project was originally started by Sun Microsystems, but its development was kept open from the very beginning. JXTA comprises six protocols allowing the discovery, organization, monitoring and communication between peers. These protocols are all implemented on the basis of an underlying messaging layer, which binds the JXTA protocols to different network transports.

FIPA has acknowledged the growing importance of the JXTA protocols, and it has released some specifications for the interoperability of FIPA platforms connected to peer-to-peer networks. In particular, in (FIPA, 2004) a generic discovery service (GDS) is described, to discover agents and services deployed on FIPA platforms working together in a peer-to-peer network. RAVE integrates a JXTA-based agent discovery service (ADS), which has been developed in the respect of relevant FIPA specifications to implement a GDS. This way, each RAVE platform connects to the Agent Peer Group, as well as to other system-specific peer groups. The generic discovery protocol is finally used to advertise and discover agent descriptions, wrapped in generic discovery advertisements, in order to implement a DF service, which in the background is spanned over a whole peer group.

CONCLUSION

In these last years, a lot of work has been done towards the realization of systems supporting virtual communities. One of the main problems in the realization of such a system is that virtual communities can be realized for a large number of purposes (i.e., exchange of information on different topics and collaboration in different work areas); therefore, an easy customization to different purposes is an important key of success of any software tool or system that is proposed for realizing solution for different types of virtual communities. RAVE is one of such systems because it is able to provide services for the virtual communities offering information and expert searching facilities for people working or interested in common or similar topics. In fact, each RAVE “customization” maintains the same agent infrastructure changing or extending the information and users profiling algorithms and if necessary a specialized user interface.

Challenges for the evolution of such of systems could be oriented to: 1) provide the automatic execution of computer-based programs offered by the different
members of a community (e.g., automatically answer with the result to the following request: “May somebody extract the text from this image?”), for example, by using Web services technologies; 2) provide tools to define multi-users flows of interactions (e.g., when member Z send me a Document Y on the topic X, ask member W to evaluate Y and send me the evaluation), for example, by using workflow technologies; and 3) to reduce the ambiguity that can raise during the interaction of two members of a community (e.g., members use different terms for indicating the same concept), for example, by using ontology and semantic Web technologies.

The previous three challenges are also guiding the work to extend RAVE. In fact, in the near future RAVE should be able to cope with the provision of a set of services coping with such challenges, but maintaining the important property of requiring few work to customize it for different applications domain. After these enhancements, an updated and detailed analysis of the key features of RAVE, compared with similar systems, will be necessary. The development will be helped by the use of multi-agent technologies and, in particular, by using the JADE agent development environment that already provided supports for Web services integration and ontology management.

REFERENCES


KEY TERMS

Information Management: Information management is the entire process dedicated to the handling of information acquired by one or many disparate sources in a way that optimizes access by all who have a share in that information or a right to that information.

Multi-agent System: A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each software agent.

Peer-to-peer Network: A peer-to-peer network is a type of computer network in which each node has equivalent capabilities and responsibilities; that is, a computer network where each node is able to simultaneously function as both “clients” and “servers” to the other nodes on the network.

Recommendation System: A Recommendation system is a program which attempts to predict items (documents, Web pages, persons, etc.) that a user may be interested in, given some information about the items and/or the user’s profile.

Social Software System: A social software system enables people to rendezvous, connect or collaborate through computer-mediated communication and to form virtual communities.

Software Agent: A software agent is a computer program that is situated in some environment and capable of autonomous action in order to meet its design objectives.

Virtual Community: A virtual community is a group of people communicating or interacting with each other by using computer networks rather than in person.